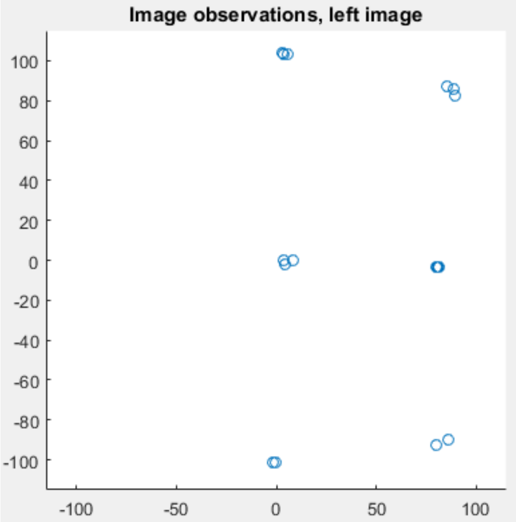
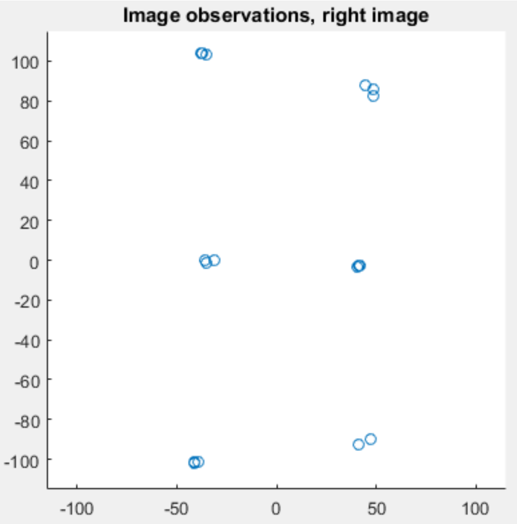
# GIS-E3010 Least-Squares Methods in Geosciences

## Photogrammetric assignment

In this assignment we practice least squares method with several photogrammetric tasks. We have two simulated aerial images with image observations and ground control points. With these observations, your task is to compute (programming with Matlab) the exterior orientation of one of the images, the relative orientation of two images and the coordinates of object points with space intersection. The simulated images (size 23 x 23 cm) have ca. 80 % forward overlap and there are 18 image measurements per image (distributed almost like Von Gruber point positions, 3 observations per point). All observations are ideal and no image corrections are needed. All initial values of unknown parameters are given. In reality, these approximate values should be computed by utilizing direct solution methods.





In addition, the principle of bundle block adjustment is practiced with a pen and paper. This part doesn't include any programming.

## Evaluation

The assignment is evaluated. The weight of the photogrammetric assignments to the final grade is 1/3 of all assignments. You can have 90 points which is scaled to the same scale with other assignments of the course. Include in your report also feedback about this assignment and estimate how much you needed time to complete it.

**Deadline** is 15.2.2019.

## Requirements

Programming parts require Matlab with Symbolic Math Toolbox, which is available in Aalto computers.

## Help

There are two exercise support meetings in which the programming parts of the assignment are assisted (Wed 30.1. 14-16 and Wed 6.2. 14-16).

If you have any further questions: petri.ronnholm@aalto.fi

## Solving exterior orientation (27 p)

In this assignment we assume that object points are constants i.e. extremely accurate points.

The aim is to find unknown parameters of collinearity equations (*ω, φ, κ, $X_0$, $Y_0$, $Z_0$*). The upper equation is referred with $f_x$ and the lower one with $f_y$.

$$\begin{cases} x = -c\dfrac{r_{11}(X-X_0)+r_{12}(Y-Y_0)+r_{13}(Z-Z_0)}{r_{31}(X-X_0)+r_{32}(Y-Y_0)+r_{33}(Z-Z_0)} = \dfrac{U}{W} = f_x \\ y = -c\dfrac{r_{21}(X-X_0)+r_{22}(Y-Y_0)+r_{23}(Z-Z_0)}{r_{31}(X-X_0)+r_{32}(Y-Y_0)+r_{33}(Z-Z_0)} = \dfrac{V}{W} = f_y \end{cases}$$

The 3D rotation matrix in this equation is

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = R_\kappa R_\varphi R_\omega = \begin{bmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix}$$

$$= \begin{bmatrix} \cos\varphi\cos\kappa & \cos\omega\sin\kappa + \sin\omega\sin\varphi\cos\kappa & \sin\omega\sin\kappa - \cos\omega\sin\varphi\cos\kappa \\ -\cos\varphi\sin\kappa & \cos\omega\cos\kappa - \sin\omega\sin\varphi\sin\kappa & \sin\omega\cos\kappa + \cos\omega\sin\varphi\sin\kappa \\ \sin\varphi & -\sin\omega\cos\varphi & \cos\omega\cos\varphi \end{bmatrix}$$

The structure of equations and trigonometric functions makes it non-linear, and therefore it should be linearized. We are trying to solve improvements (=corrections) to initial values of parameters (or to values from the previous iteration round) with least squares method. The values of unknown parameters are corrected (by summing corrections to current values) for the next iteration round. Iteration can be ended if there are no more significant improvements to unknown parameters.

Let's move to the details

Answer also to the questions that are given

**Q**: What are the correspondence observations in this case? (1 p)

**Q**: How many equations one corresponding observation establish? (1 p)

**Q**: How many equations six corresponding observations establish? (1 p)

**Q**: How many unknown parameters you have? (1 p)

**Q**: How many equations you need to establish (minimum) in order to solve unknowns and how many corresponding observations you need then? (1 p)

Note: it is advantageous to have more corresponding observations than minimum.

**Q**: What is the redundancy of the system if we have 4 corresponding observations? (1 p)

The next phase is to create the error equation

$$v = Ax - y$$

For one corresponding point measurement this looks like

$$\begin{bmatrix} v_x \\ \\ v_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial \omega} & \frac{\partial f_x}{\partial \varphi} & \frac{\partial f_x}{\partial \kappa} & \frac{\partial f_x}{\partial X_0} & \frac{\partial f_x}{\partial Y_0} & \frac{\partial f_x}{\partial Z_0} \\ \\ \frac{\partial f_y}{\partial \omega} & \frac{\partial f_y}{\partial \varphi} & \frac{\partial f_y}{\partial \kappa} & \frac{\partial f_y}{\partial X_0} & \frac{\partial f_y}{\partial Y_0} & \frac{\partial f_y}{\partial Z_0} \end{bmatrix} \begin{bmatrix} d\omega \\ d\varphi \\ d\kappa \\ dX_0 \\ dY_0 \\ dZ_0 \end{bmatrix} - \begin{bmatrix} x - f_x^0 \\ \\ y - f_y^0 \end{bmatrix}$$

We are going to solve the correction vector of unknown parameters *x*. For that we need to establish everything else (*A* and *y*).

**Q**: Write this equation for 4 corresponding points using general symbols (add sub-index to x and y indicating the number of corresponding observation), e.g. for the first corresponding point (1 p)

$$\begin{bmatrix} v_{x1} \\ \\ v_{y1} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_{x1}}{\partial \omega} & \frac{\partial f_{x1}}{\partial \varphi} & \frac{\partial f_{x1}}{\partial \kappa} & \frac{\partial f_{x1}}{\partial X_0} & \frac{\partial f_{x1}}{\partial Y_0} & \frac{\partial f_{x1}}{\partial Z_0} \\ \\ \frac{\partial f_{y1}}{\partial \omega} & \frac{\partial f_{y1}}{\partial \varphi} & \frac{\partial f_{y1}}{\partial \kappa} & \frac{\partial f_{y1}}{\partial X_0} & \frac{\partial f_{y1}}{\partial Y_0} & \frac{\partial f_{y1}}{\partial Z_0} \end{bmatrix} \begin{bmatrix} d\omega \\ d\varphi \\ d\kappa \\ dX_0 \\ dY_0 \\ dZ_0 \end{bmatrix} - \begin{bmatrix} x_1 - f_{x1}^0 \\ \\ y_1 - f_{y1}^0 \end{bmatrix}$$

In practice, only vectors *v* and *y,* and matrix *A* grow (they get more rows).

The design matrix *A* is a Jacobian matrix in which the equations are partially derivated with respect to all unknown parameters. Each column is reserved for one unknown parameter.

**Matlab task:** create partial derivatives of the design matrix *A*. Add you code to the report. (10 p)

Even if these partial derivatives can be found from literature (and can be done by pen and paper), use the symbolic library of Matlab to create your version of them. Instructions are given in Appendix 1.

$$\begin{bmatrix} v_x \\ \\ v_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial \omega} & \frac{\partial f_x}{\partial \varphi} & \frac{\partial f_x}{\partial \kappa} & \frac{\partial f_x}{\partial X_0} & \frac{\partial f_x}{\partial Y_0} & \frac{\partial f_x}{\partial Z_0} \\ \\ \frac{\partial f_y}{\partial \omega} & \frac{\partial f_y}{\partial \varphi} & \frac{\partial f_y}{\partial \kappa} & \frac{\partial f_y}{\partial X_0} & \frac{\partial f_y}{\partial Y_0} & \frac{\partial f_y}{\partial Z_0} \end{bmatrix} \begin{bmatrix} d\omega \\ d\varphi \\ d\kappa \\ dX_0 \\ dY_0 \\ dZ_0 \end{bmatrix} - \begin{bmatrix} x - f_x^0 \\ \\ y - f_y^0 \end{bmatrix} \qquad v = Ax - y$$

The next step is to understand a vector *y*, which includes the difference between observed image coordinates (*x*,*y*) and estimated image observations ( $f_x^0, f_y^0$ ). Estimated image observations are computed with initial values of parameters i.e. you use collinearity equations for each ground control points to compute their projection to the image plane (estimated location) with approximate orientation parameters. You can check if your partial derivatives are correct by copying the result into function check_partial_derivatives.m, and by running it.

**Matlab task**: Complete given Matlab function, establish the *A* matrix and the *y* vector, and compute the exterior orientation parameters. Add your code and results in your report. (10 p)

## Solving relative orientation (26 p)

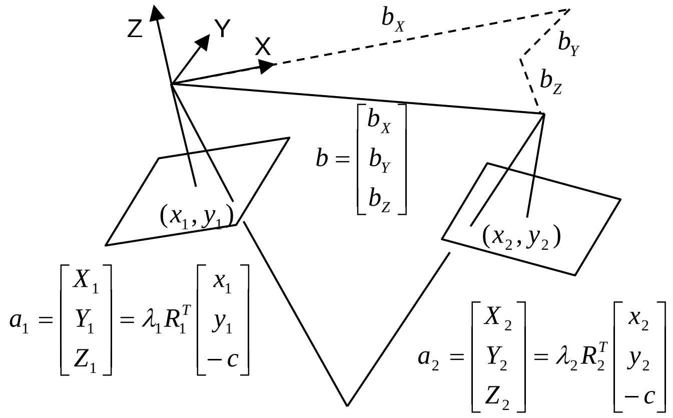
In this case the mathematical model is the coplanarity equation and we want to solve the relative orientation of successive images (i.e. one image remains stable and the other one is moved and rotated).

$$G = \begin{vmatrix} b_X & b_Y & b_Z \\ X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \end{vmatrix} = b_X(Y_1Z_2 - Y_2Z_1) - b_Y(X_1Z_2 - X_2Z_1) + b_Z(X_1Y_2 - X_2Y_1) = 0$$

in which

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = R_1^T \begin{bmatrix} x_1 \\ y_1 \\ -c \end{bmatrix}$$

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = R_2^T \begin{bmatrix} x_2 \\ y_2 \\ -c \end{bmatrix}$$


Z
Y
X
$b_X$
$b_Y$
$b_Z$
$b = \begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix}$
$(x_1, y_1)$
$(x_2, y_2)$
$a_1 = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \lambda_1 R_1^T \begin{bmatrix} x_1 \\ y_1 \\ -c \end{bmatrix}$
$a_2 = \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \lambda_2 R_2^T \begin{bmatrix} x_2 \\ y_2 \\ -c \end{bmatrix}$

The interpretation of rotation matrix is the same than previously (in the case of exterior orientation). Because of the presence of trigonometric functions, the equation is non-linear and needs to be linearized. As usually, we need to have some initial values for all unknown parameters that are relatively close to the correct ones (initial values are given in this assignment). We use mixed model and general least squares adjustment. The structure is

$$\begin{bmatrix} \frac{\partial G^0}{\partial x_1} & \frac{\partial G^0}{\partial y_1} & \frac{\partial G^0}{\partial x_2} & \frac{\partial G^0}{\partial y_2} \end{bmatrix} \begin{bmatrix} dx_1 \\ dy_1 \\ dx_2 \\ dy_2 \end{bmatrix} + \begin{bmatrix} \frac{\partial G^0}{\partial b_Y} & \frac{\partial G^0}{\partial b_Z} & \frac{\partial G^0}{\partial \omega_2} & \frac{\partial G^0}{\partial \varphi_2} & \frac{\partial G^0}{\partial \kappa_2} \end{bmatrix} \begin{bmatrix} db_Y \\ db_Z \\ d\omega_2 \\ d\varphi_2 \\ d\kappa_2 \end{bmatrix} + G^0 = 0$$

i.e. $C^T dl + Ddp + G^0 = 0$

**Q**: What are the correspondence observations in this case? (1 p)

**Q**: How many equations one corresponding observation establish? (1 p)

**Q**: For how many parameters we solve corrections? (2 p)

**Q**: What is the redundancy of the system if we have 9 corresponding observations? (1 p)

**Q**: How can you define the scale of the resulting 3D model? (1 p)

**Matlab task**: create partial derivatives of the matrices C and D (template make_all_partial_derivates_C_and_D_matrices_RO.m). (notice that this part is already solved in order to reduce the workload i.e. you get free 10 points. However, check the code and try to understand what has been done. If you have too much free time, you may try to solve this by yourself). (10 p)

Copy the results (they will be saved in the file partial_derivatives_RO.m) to the Matlab template solve_RO_ls.m

**Matlab task**: Complete the template solve_RO_ls.m. Add you code and results to the report. (10 p)

## Solve spatial intersection (12 p)

In this case we know exterior orientations of two images. According to image observation we aim to make space intersection of observation vectors and get 3D object coordinates. For this task, we use collinearity equations.

$$\begin{cases} x = -c\dfrac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} = f_x \\ y = -c\dfrac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} = f_y \end{cases}$$

If these equations are developed a bit we get

$$\begin{cases} (xr_{31} + cr_{11})X + (xr_{32} + cr_{12})Y + (xr_{33} + cr_{13})Z = (xr_{31} + cr_{11})X_0 + (xr_{32} + cr_{12})Y_0 + (xr_{33} + cr_{13})Z_0 \\ (yr_{31} + cr_{21})X + (yr_{32} + cr_{22})Y + (yr_{33} + cr_{23})Z = (yr_{31} + cr_{21})X_0 + (yr_{32} + cr_{22})Y_0 + (yr_{33} + cr_{23})Z_0 \end{cases}$$

In this equation, all unknown parameters are on the left side of the “=” sign and all scalars are on the right side of “=” sign. Because we know rotations, this equation is linear and we get a direct solution (no initial values are needed, nor iteration).

**Q**: What are the unknown parameters? (1 p)

The design matrix (A) is a Jacobian matrix (as in the two previous cases) and y contains the observations (the right side of the equation). $v = Ax - y$

**Q**: Write the contents of A matrix with symbols like in the case of exterior orientation for one corresponding observation pair (one point from the left image and one from the right image)

(use $f_{x1}$, $f_{y1}$, $f_{x2}$, $f_{y2}$ to separate left (1) and right (2) image observations)) (1 p)

Reminder: Example from the case of exterior orientation: $A = \begin{bmatrix} \frac{\partial f_x}{\partial \omega} & \frac{\partial f_x}{\partial \varphi} & \frac{\partial f_x}{\partial \kappa} & \frac{\partial f_x}{\partial X_0} & \frac{\partial f_x}{\partial Y_0} & \frac{\partial f_x}{\partial Z_0} \\ \frac{\partial f_y}{\partial \omega} & \frac{\partial f_y}{\partial \varphi} & \frac{\partial f_y}{\partial \kappa} & \frac{\partial f_y}{\partial X_0} & \frac{\partial f_y}{\partial Y_0} & \frac{\partial f_y}{\partial Z_0} \end{bmatrix}$

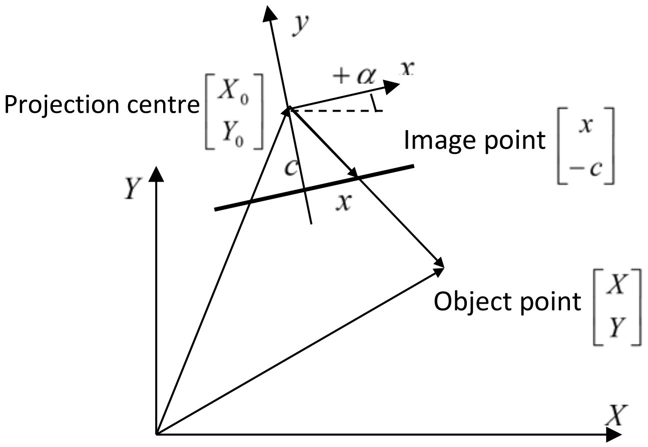
**Q**: What is the redundancy of the system? (1 p)

**Matlab task**: Complete the Matlab template space_intersection.m. Add you code and results to the report. (9 p)

## Bundle block adjustment (25 p)

In this case we examine bundle block adjustment when

- The camera is 1-dimensional (a line camera) and
- The object space is 2-dimensional (an XY plane)



$y$
$+\alpha$
$x$
Projection centre $\begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$
Image point $\begin{bmatrix} x \\ -c \end{bmatrix}$
$c$
$x$
$Y$
Object point $\begin{bmatrix} X \\ Y \end{bmatrix}$
$X$

Figure 1

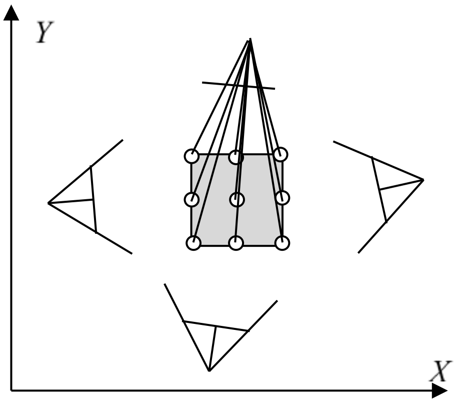


Figure 2

In this case (1D image and 2D object space), the collinearity equations become as:

$$\begin{bmatrix} x \\ -c \end{bmatrix} = k \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} X - X_0 \\ Y - Y_0 \end{bmatrix} = kR \begin{bmatrix} X - X_0 \\ Y - Y_0 \end{bmatrix} \text{ i.e.}$$

$$x = -c \frac{(X - X_0)\cos\alpha + (Y - Y_0)\sin\alpha}{-(X - X_0)\sin\alpha + (Y - Y_0)\cos\alpha} = -c\frac{U}{W}$$

**Q**: In this case, for which transformation (of the object space) the image observations are invariants? (1 p)

**Task**: Make linearization (partial derivatives with respect to all unknown parameters) of this collinearity equation using 1st order Taylor series (either with a pen and paper of with the Matlab symbolic library) (5 p)

$$x = x^0 + \frac{\partial x^0}{\partial \alpha} d\alpha + \frac{\partial x^0}{\partial X_0} dX_0 + \frac{\partial x^0}{\partial Y_0} dY_0 + \frac{\partial x^0}{\partial X} dX + \frac{\partial x^0}{\partial Y} dY$$

(Calculate all partial derivatives, such as $\partial x / \partial_{X_0} \alpha$ etc.). When using a pen and paper approach, use the rule of partial derivation of quotient equation: $\frac{\partial x}{\partial p_k} = -c\left(\frac{\partial U}{\partial p_k} W - \frac{\partial W}{\partial p_k} U\right) / W^2 = \frac{-c}{W}\left(\frac{\partial U}{\partial p_k} - \frac{U}{W}\frac{\partial W}{\partial p_k}\right)$

**Q**: How many parameters we need in order to define the *exterior orientation* of a single image (in this special case)? (1 p)

**Q**: How many object points we need to know in order to solve exterior orientation parameters? (1 p)

**Q**: What is (a) *the number of observations*, (b) *the number of unknown parameters* and (c) *redundancy* if we have a bundle block adjustment that includes four images and nine object points (see Figure 2). We assume that all object points are unknown but visible in all images. (3 p)

**Q**: What is the amount of *datum defect* in the case that was described in the previous question? And what is the *rank* of design matrix? (2 p)

**Q**: The simplest method how to remove datum defect is to ”fix” some of the object points. How many points/coordinates we have to “fix” in this case? Write out the required additional error equations (outer constraints) of object coordinates $E^T x=0$. (4 p)

**Q**. Datum defect can be removed also by using *inner constraints*. Write out the additional error equations $E^T x=0$ in this case. (For the simplicity, we take account only object coordinates – not exterior orientations). Use the following equation to derive the result: (3 p)

$$\begin{bmatrix} dX \\ dY \end{bmatrix} = \begin{bmatrix} dX_0 \\ dY_0 \end{bmatrix} + \begin{bmatrix} 0 & d\alpha \\ -d\alpha & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + d\lambda \begin{bmatrix} X \\ Y \end{bmatrix}$$

**Task**. Write out the structure of (free network) bundle block adjustment (a matrix A and vectors $\Delta$ and f) using symbols in the case that includes two images and four object points. A free network adjustment requires either inner or outer constraints in order to remove datum defect, and you already solved those in questions 8 and 9. Now you don’t have to make a solution of equations, but we would get it from (5 p)

$$\begin{bmatrix} N & E \\ E^T & 0 \end{bmatrix} \begin{bmatrix} \Delta \\ k \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix}$$ in which $$N = A^T PA \qquad u = A^T Pf$$

# Appendix 1. Creating symbolic partial derivatives with respect to unknown parameters in Matlab

You have been given a template to create symbolic partial derivatives:
`make_all_partial_derivates_A_matrix.m`

This file includes several functions. The first function make_all_partial_derivates_A_matrix runs all other functions and writes the results in a new file partial_derivatives.m. Your task is to complete all sub functions, e.g.

```
function [ fx_d_o ] = derive_fx_d_o()
function [ fy_d_o ] = derive_fy_d_o()
```

etc.

Even if there are 12 sub-functions to complete the two first ones are the most important. In practice, when you have succeeded to make these two functions the rest can be mostly copy-pasted from them and just change some symbols.

In Matlab, symbolic variables are initialized with “syms”. You need to do this in each function.  If for example we use o=omega, p=phi and k=kappa, we could initialize these symbols

```
syms o p k;
```

Initialize symbols omega, phi, kappa, X0, Y0, Z0, c, X, Y, Z, r11, r21, r31, r21, r22, r23, r31, r32, r33, which are needed for collinearity equations. Start with establishing the elements of 3D rotation matrix, because then it is easier to create collinearity equation. For example, the first element is

```
r11=cos(p)*cos(k);
```

The next step is to write collinearity equation. In derive_fx_d_o (and all with fx) you need to create only the upper equation of the collinearity equations.

fx=-c*(r11*…

This equation is partially derivated with respect to current unknown. For example, in derive_fx_d_o this variable is omega. Use Matlab function diff() to make this derivation and place the result in the outgoing parameter of the function, e.g. in

```
function [ fx_d_o ] = derive_fx_d_o()
%fx_d_o is the outgoing parameter and therefore

fx_d_o=diff(…
```

After you have managed to complete all functions run make_all_partial_derivates_A_matrix ▷ Run. If you open the resulting file “partial_derivatives.m”, you can see several functions in which the results from the previous step are collected as Matlab-functions. You may check if everything is in condition. Copy all functions from “partial_derivatives.m” and copy them in to the file “check_partial_derivatives.m”. When you run this function, it tells if you have everything in condition or if you have problems in some sub-functions (only for the case of exterior orientation).